

API

ООО «Электронные системы и
сервисы»

[API СЕРВИСА «ХУТКІ ГРОШ™»]

Документация по внешним интерфейсам сервиса «Хуткі Грош™»

Дата	Версия	Правил	Комментарий
19.12.2010	draft	Прокша К.Э.	
20.01.2011	1.0	Прокша К.Э.	
15.04.2011	1.1	Прокша К.Э.	SOAP сервис
09.06.2011	1.2	Прокша К.Э.	Примеры кода
21.07.2011	1.3	Прокша К.Э.	Коды ошибок
11.08.2011	1.4	Прокша К.Э.	Примеры на PHP
19.09.2011	1.5	Прокша К.Э.	Дополнительные коды ошибок, описание нотификации
29.09.2011	1.6	Прокша К.Э.	Изменение структуры Bill, добавлено поле firstPaymentAmt
02.10.2011	1.7	Прокша К.Э.	Добавлено описание подсистемы работы с курсами валют
03.10.2011	1.8	Прокша К.Э.	Изменение структуры Bill, добавлено поле paymentSource
20.02.2012	1.9	Прокша К.Э.	Добавлены новые методы BillsEx и BillsInMemorialSlip. И новый метод SOAP сервиса EP_VerifyPaidInvoices.
21.02.2012	1.10	Прокша К.Э.	Исправлены неточности в примерах.
25.09.2014	1.11	Прокша К.Э.	Добавлен метод подготовки формы оплаты карточками VISA/Master. Подсистема оплаты.
12.03.2019	1.12	Аскерко А.В.	Добавление метода получения данных по qr-коду счета
29.03.2022	1.13	Аскерко А.В.	Добавление идентификатора операции ЕРИП в структуру счета

Как использовать API	4
Нотификация	5
Важные замечания	6
Примеры кода	7
Документация	9
Подсистема аутентификации	9
Метод LogIn	9
Метод LogOut	10
Подсистема работы со счетами	11
Метод Bill	11
Метод Bill({billID})	15
Метод BillQRCode({billID}, {RESULTTYPE}, {STRWIDTH}, {STRHEIGHT})	17
Метод Bill({billID})	19
Метод Bills({startPosition},{requestItemsCount},{sortType})	20
Метод BillsEx({startDt},{endDt},{startPosition},{requestItemsCount}, {sortType},{spName},{purchItemStatus})	24
Метод BillStatus({billID})	27
Метод PayedBills({eripId},{order})	29
Метод BillsInMemorialSlip({memSlip})	32
Подсистема оплаты	35
Метод webPay	35
Метод AlfaClick	36
Подсистема Работы с курсами валют	37
Метод Currency({isArchive})	37
Метод Currency	39
SOAP сервис	42
Метод EP_GetPaidInvoicesExt	42
Метод EP_VerifyPaidInvoices	43
Структуры данных и перечисления	44
Структуры	44

Перечисления	47
Коды ошибок.....	49

Все вызовы методов API – это GET, POST, DELETE HTTP-запросы к URL. Базовый адрес для вызова методов:

<https://www.hutkigrosh.by/API/v1/<Подсистема>/<Метод>>

с некоторым набором параметров. Для выполнения какого-либо действия, необходимо сформировать запрос, согласно документации вызываемого метода, и осуществить этот запрос к серверу. В ответ на запрос вы получаете его результат, который также описан в документации каждого метода. Кодировка результата UTF-8. Работа с API возможна как в формате Json, так и XML. Зависит от заданного content-type в запросе. Нужно указывать "application/json" для Json формата или "application/xml" для XML. В запросе важен порядок прописных и строчных букв.

Для осуществления запросов необходимо пройти аутентификацию. Аутентификация выполняется с помощью вызова специального метода:

<https://www.hutkigrosh.by/API/v1/Security/LogIn>

Полученный набор cookie, после вызова этого метода, необходимо использовать в последующих вызовах других методов. По окончании работы с сервисом, необходимо завершить сеанс вызовом метода:

<https://www.hutkigrosh.by/API/v1/Security/LogOut>

Примечание:

Для тестирования API необходимо использовать тестовую площадку **trial.hgrosh.by**, сертификат, установленный на тестовой площадке, является самоподписанным, так что для некоторых фреймворков необходимо отключать проверку валидности сертификата сервера. Регистрационные данные, для проведения тестирования, выдаются техническими специалистами сервиса «ХУТКИ ГРОШ™» по запросу.

Сервис "Хуткі Грош" может извещать Поставщика услуг тремя способами:

1. С помощью SMS сообщений;
2. Путем отправки E-Mail;
3. Вызовом специального URL, указанного Поставщиком услуг.

Любое извещение формируется в момент совершения платежа физическим лицом. Как только транзакция завершена, и деньги списаны с клиента, наш сервис генерирует нотификационное событие.

Первые два способа прозрачны и не требуют дополнительного разъяснения. Разберем третий способ нотификации. Этот способ позволяет снизить нагрузку на сервера, так как нет необходимости постоянно опрашивать на предмет появления новых оплаченных счетов. В момент оплаты, наш сервис вызовет URL вида:

[http\[s\]://domain.zone/path/?purchaseid={billId}](http[s]://domain.zone/path/?purchaseid={billId})

где, [http\[s\]://domain.zone/path/](http[s]://domain.zone/path/) - путь к Вашему сервису, принимающему уведомления, {billId} – это наш внутренний идентификатор оплачиваемого счета. Вы должны его использовать, чтоб запросить детальную информацию об оплачиваемом счете, например, вызывая метод [https://www.hutkigrosh.by/API/v1/Invoicing/Bill\({billId}\)](https://www.hutkigrosh.by/API/v1/Invoicing/Bill({billId})). Ваш сервис должен ответить кодом 200, если он обработал данное сообщение или любым другим, если произошла ошибка. В случае, когда мы получили код, отличный от 200, наш сервис будет повторять уведомление еще 3 раза с нарастающим интервалом времени. Если все 4 попытки оказались неудачными, то высылается уведомление на указанный Вами email, в котором отправляется внутренний идентификатор оплаченного счета и лог попыток уведомления.

ВАЖНЫЕ ЗАМЕЧАНИЯ

Для получения ответа от сервера в формате **XML**, то необходимо в заголовке запроса указать:

Accept: application/xml

Для получения ответа в формате **Json**, то необходимо в заголовке запроса указать:

Accept: application/json

За формат запроса отвечает поле:

Content-Type

Дата представлена в **Json** запросах в **.NET** нотации `"/Date(123456789000)/"`, где цифры, это тики в миллисекундах от 01.01.1970. Например `"/Date(1000)/"` – это 01.01.1970 00:00:01.

При выставлении счета необходимо следить, что бы поле **dueDt** было в будущем периоде, иначе будет возвращена общая ошибка выставления счета.

Код, приведенный в данной документации, написан на C# с использованием .NET Framework и на PHP с использованием библиотеки PECL_HTTP (http://pecl.php.net/package/pecl_http).

В примерах, приведенных в документации, для упрощения кода в части выполнения HTTP запросов и получения ответов на них используется функция, примерная реализация которой приведена ниже:

C#

```
public static string Request( string URL, string method, string contentType,
                             string strRequest, ref CookieContainer cookie )
{
    WebResponse rs = null;
    HttpWebRequest rq = null;
    Stream rsStream = null;
    StreamReader sr = null;
    string strOut = "";

    try
    {
        rq = ( HttpWebRequest ) WebRequest.Create( URL );
        rq.CookieContainer = cookie;
        rq.Method = method;
        rq.Timeout = 120000;
        rq.ContentType = contentType;

        if( !string.IsNullOrEmpty( strRequest ) )
        {
            byte[] rqData = null;
            rqData = Encoding.GetEncoding( "UTF-8" ).GetBytes( strRequest );
            rq.ContentLength = rqData.Length;
            using( Stream rqStream = rq.GetRequestStream() )
            {
                rqStream.Write( rqData, 0, rqData.Length );
                rqStream.Close();
            }
        }
        rs = rq.GetResponse();
        rsStream = rs.GetResponseStream();
        Encoding encode = Encoding.GetEncoding( "UTF-8" );
        sr = new StreamReader( rsStream, encode );

        Char[] read = new Char[ 256 ];
        int count = sr.Read( read, 0, 256 );

        while( count > 0 )
        {
            String str = new String( read, 0, count );
            strOut += str;
            count = sr.Read( read, 0, 256 );
        }
    }
    catch( WebException ex )
    {
        strOut = null;
    }
    catch( Exception ex )
    {
        strOut = null;
    }
    finally
    {
        if( rsStream != null )
            rsStream.Close();
    }
}
```



```
    if( sr != null )
        sr.Close();
    if( rs != null )
        rs.Close();
}
return strOut;
}
```

PHP

```
function sendToHG( $req, $url, $data, $cookies, $method )
{
    $req -> setUrl( $url );
    $req -> setHeaders(
        array(
            "Content-Type" => "application/xml",
            "Content-Length" => strlen( $data )
        )
    );
    $req -> setMethod( $method );
    $req -> setRawPostData( $data );
    $req -> enableCookies();
    if( $cookies != null )
        $req->setCookies( $cookies );
    $req -> send();
    return $req -> getResponseBody();
}
```

Так же для простоты все запросы будут сформированы в виде текстовых строк.

ПОДСИСТЕМА АУТЕНТИФИКАЦИИ

<https://www.hutkigrosh.by/API/v1/Security>

Метод		Описание
LogIn	POST	Аутентифицирует пользователя в системе
LogOut	POST	Завершает сессию

МЕТОД LOGIN

HTTP verb: **POST**

ОПИСАНИЕ

Аутентифицирует пользователя в системе.

ВЫЗОВ

<https://www.hutkigrosh.by/API/v1/Security/LogIn>

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

Для вызова данного метода необходимо сформировать тело запроса. Тело запроса зависит от выбранного значения **content-type**:

XML

```
<Credentials xmlns="http://www.hutkigrosh.by/api/">
  <user>username@org.com</user>
  <pwd>pSSw_ord7</pwd>
</Credentials>
```

Json

```
{
  "user": "username@org.com",
  "pwd": "pSSw_ord7"
}
```

ОТВЕТ

Ответ приходит **true** – если аутентификация успешна, **false** – если проверка имени и пароля провалилась.

XML

```
<boolean xmlns="http://schemas.microsoft.com/2003/10/Serialization/">true</boolean>
```

Json

```
true
```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rd</pwd>" +
    "</Credentials>";
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );
```

PHP

```
$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rd</pwd>".
    "</Credentials>";

echo "Login : ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> getCookies();
```

МЕТОД LOGOUT

HTTP verb: **POST**

ОПИСАНИЕ

Метод завершает сессию.

ВЫЗОВ

<https://www.hutkigrosh.by/API/v1/Security/Logout>

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

Нет параметров.

ОТВЕТ

Нет возвращаемых данных.

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом и завершение сеанса.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rd</pwd>" +
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
```

```

// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
                    "POST", "application/xml", creds, ref authCookie );

//
// Ваши вызовы в данном сеансе помещать здесь
//

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
              "POST", "application/xml", "", ref authCookie );

```

PHP

```

$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns='http://www.hutkigrosh.by/api/'>".
           "<user>user@org.com</user>".
           "<pwd>paSSwo_rd</pwd>".
           "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login : ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

//
// Ваши вызовы в данном сеансе помещать здесь
//

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/LogOut";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";

```

ПОДСИСТЕМА РАБОТЫ СО СЧЕТАМИ

<https://www.hutkigrosh.by/API/v1/Invoicing>

Метод		Описание
Bill	POST	Добавляет новый счет в систему
Bill({billID})	GET	Извлекает информацию о выставленном счете
	DELETE	Удаляет выставленный счет из системы
Bills({startPosition},{requestItemsCount},{sortType})	GET	Возвращает список выставленных счетов
BillStatus({billID})	GET	Возвращает статус указанного счета
PayedBills({eripId},{order})	GET	Возвращает список оплаченных счетов

МЕТОД BILL

HTTP verb: **POST**

ОПИСАНИЕ

Добавляет новый счет в систему.

ВЫЗОВ

<https://www.hutkigrosh.by/API/v1/Invoicing/Bill>

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

Для вызова данного метода необходимо сформировать тело запроса. Запрос представляет собой заполненную структуру **Bill**, описание полей структуры находится в разделе "Структуры данных и перечисления". Тело запроса зависит от выбранного значения **content-type**:

XML

```
<Bill xmlns="http://www.hutkigrosh.by/api/invoicing">
  <eripId>40000001</eripId>
  <invId>C-1234</invId>
  <dueDt>2011-04-18T11:20:00</dueDt>
  <addedDt>2011-04-18T11:20:00</addedDt>
  <fullName>Пупкин Василий Иванович</fullName>
  <mobilePhone>+333 33 3332221</mobilePhone>
  <notifyByMobilePhone>true</notifyByMobilePhone>
  <email /> <!--не включать, если не заполняется-->
  <notifyByEmail>false</notifyByEmail>
  <fullAddress>г.Минск, пр.Победителей, д.1, кв.1</fullAddress>
  <amt>120000.000000</amt>
  <curr>BYN</curr>
  <statusEnum>NotSet</statusEnum>
  <info /> <!--не включать, если не заполняется-->
  <products>
    <ProductInfo>
      <invItemId>Артикул 123</invItemId>
      <desc>Услуга, за которую производят оплату</desc>
      <count>1</count>
      <amt>119000.000000</amt>
    </ProductInfo>
    <ProductInfo>
      <invItemId>-нет-</invItemId>
      <desc>Доставка</desc>
      <count>1</count>
      <amt>1000.000000</amt>
    </ProductInfo>
  </products>
</Bill>
```

Json

```
{
  "billID":null,
  "eripId":40000001,
  "invId":"C-1234",
  "dueDt":"\\/Date(928138800000+0300)\\/\"",
  "addedDt":"\\/Date(928138800000+0300)\\/\"",
  "payedDt":null,
  "fullName":"Пупкин Василий Иванович",
  "mobilePhone":"+333 33 3332221",
  "notifyByMobilePhone":true,
  "email":null,
  "notifyByEmail":false,
  "fullAddress":"г.Минск, пр.Победителей, д.1, кв.1",
  "amt":120000.000000,
  "curr":"BYN",
  "statusEnum":0,      "info":null,
  "products":[{"
    "invItemId":"Артикул 123",
    "desc":"Услуга, за которую производят оплату",
    "count":1,
    "amt":120000.000000
  }]
}
```

ОТВЕТ

Ответом является структура **NewBillStatus**, в которой проставлены статус выполнения запроса и, в случае успешного завершения запроса и выставления счета в систему, внутренний идентификатор счета.

XML

```
<NewBillStatus xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>0</status>
  <billID>4000000528202700</billID>
</NewBillStatus>
```

Json

```
{
  "status":0,
  "billID":4000000528202700
}
```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, выставление нового счета в систему и завершение сеанса.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">\" +
               "<user>user@org.com</user>\" +
               "<pwd>paSSwo_rdc</pwd>\" +
               "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
                    "POST", "application/xml", creds, ref authCookie );

// Выставляем новый счет
res = "<Bill xmlns=\"http://www.hutkigrosh.by/api/invoicing\">\" +
      //\"<billID />\" +
      "<eripId>40000001</eripId>\" +
      "<invId>C-1234</invId>\" +
      "<dueDt>2011-04-18T11:20:00</dueDt>\" +
      "<addedDt>2011-04-18T11:20:00</addedDt>\" +
      //\"<payedDt />\" +
      "<fullName>Пупкин Василий Иванович</fullName>\" +
      "<mobilePhone>+333 33 3332221</mobilePhone>\" +
      "<notifyByMobilePhone>true</notifyByMobilePhone>\" +
      //\"<email />\" +
      "<notifyByEmail>false</notifyByEmail>\" +
      "<fullAddress>г.Минск, пр.Победителей, д.1, кв.1</fullAddress>\" +
      "<amt>120000.000000</amt>\" +
      "<curr>BYN</curr>\" +
      "<statusEnum>NotSet</statusEnum>\" +
      //\"<info />\" +
      "<products>\" +
      "  <ProductInfo>\" +
      "    <invItemId>Артикул 123</invItemId>\" +
      "    <desc>Услуга, за которую производят оплату</desc>\" +
      "    <count>1</count>\" +
      "    <amt>119000.000000</amt>\" +
      "  </ProductInfo>\" +
      "  <ProductInfo>\" +
      "    <invItemId>-нет-</invItemId>\" +
      "    <desc>Доставка</desc>\" +
```

```

        "<count>1</count>" +
        "<amt>1000.000000</amt>" +
        "</ProductInfo>" +
        "</products>" +
        "</Bill>";
res = Request( "https://www.hutkigrosh.by/API/v1/Invoicing/Bill", "POST",
    "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
    "POST", "application/xml", "", ref authCookie );

```

PHP

```

$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns='\"http://www.hutkigrosh.by/api\">".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rd</pwd>".
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login : ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Выставляем новый счет
$url = "https://www.hutkigrosh.by/API/v1/Invoicing/Bill";
$dataXML =
    "<Bill xmlns='\"http://www.hutkigrosh.by/api/invoicing\">".
    //"<billID />".
    "<eripId>4000001</eripId>".
    "<invId>C-1234</invId>".
    "<dueDt>2011-04-18T11:20:00</dueDt>".
    "<addedDt>2011-04-18T11:20:00</addedDt>".
    //"<payedDt />".
    "<fullName>Пупкин Василий Иванович</fullName>".
    "<mobilePhone>+333 33 3332221</mobilePhone>".
    "<notifyByMobilePhone>true</notifyByMobilePhone>".
    //"<email />".
    "<notifyByEmail>false</notifyByEmail>".
    "<fullAddress>г.Минск, пр.Победителей, д.1, кв.1</fullAddress>".
    "<amt>120000.000000</amt>".
    "<curr>BYN</curr>".
    "<statusEnum>NotSet</statusEnum>".
    //"<info />".
    "<products>".
    "<ProductInfo>".
    "<invItemId>Артикул 123</invItemId>".
    "<desc>Услуга, за которую производят оплату</desc>".
    "<count>1</count>".
    "<amt>119000.000000</amt>".
    "</ProductInfo>".
    "<ProductInfo>".
    "<invItemId>-нет-</invItemId>".
    "<desc>Доставка</desc>".
    "<count>1</count>".
    "<amt>1000.000000</amt>".
    "</ProductInfo>".
    "</products>".
    "</Bill>";
echo "Add Bill : ".sendToHG( $req, $url, $dataXML, $cookies,
    HTTP_METH_POST )."<br/>";
// Теперь необходимо обработать результат, который вернул сервер

```

```
// для этого надо изменить предыдущую строку соответствующим образом

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/LogOut";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";
```

МЕТОД BILL({BILLID})

HTTP verb: **GET**

ОПИСАНИЕ

Извлекает информацию о выставленном счете.

ВЫЗОВ

[https://www.hutkigrosh.by/API/v1/Invoicing/Bill\({billId}\)](https://www.hutkigrosh.by/API/v1/Invoicing/Bill({billId}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При вызове данного метода необходимо указать внутренний идентификатор счета, заменив параметр **{billId}** на нужное значение. Необходимо указать в запросе параметр **content-type** для определения вида возвращаемого ответа.

Пример запроса: [https://www.hutkigrosh.by/API/v1/Invoicing/Bill\(4000000528202700\)](https://www.hutkigrosh.by/API/v1/Invoicing/Bill(4000000528202700))

ОТВЕТ

Ответом является структура **BillInfo**, описание полей структуры находится в разделе "Структуры данных и перечисления".

XML

```
<BillInfo xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>0</status>
  <bill>
    <billID>4000000528202700</billID>
    <eripId>0</eripId>
    <invId>123-EEK</invId>
    <dueDt>2011-06-30T00:00:00</dueDt>
    <addedDt>2011-06-09T00:00:00</addedDt>
    <payedDt i:nil="true"/>
    <fullName>Петров Петя Петечкин</fullName>
    <mobilePhone>+375 33 3333333</mobilePhone>
    <notifyByMobilePhone>false</notifyByMobilePhone>
    <email i:nil="true"/>
    <notifyByEmail>false</notifyByEmail>
    <fullAddress i:nil="true"/>
    <amt>102000</amt>
    <curr>BYN</curr>
    <statusEnum>PaymentPending</statusEnum>
    <eripTrxId>123456789</eripTrxId>
    <info>ИООО "Ваша любимая компания"</info>
    <products>
      <ProductInfo>
        <invItemId>-</invItemId>
        <desc>За услуги связи</desc>
        <count>1</count>
        <amt>100000</amt>
      </ProductInfo>
      <ProductInfo>
        <invItemId>-</invItemId>
        <desc>За доп услуги</desc>
```



```

        <count>1</count>
        <amt>2000</amt>
    </ProductInfo>
</products>
</bill>
</BillInfo>

```

Json

```

{
  "status":0,
  "bill":{
    "billID":4000000528202700,
    "eripId":0,
    "invId":"123-EEK",
    "dueDt":"\\/Date(1309381200000+0300)\\/","
    "addedDt":"\\/Date(1307566800000+0300)\\/","
    "payedDt":null,
    "fullName":"Петров Петя Петечкин",
    "mobilePhone":"+375 33 3333333",
    "notifyByMobilePhone":false,
    "email":null,
    "notifyByEmail":false,
    "fullAddress":null,
    "amt":102000,
    "curr":"BYN",
    "statusEnum":1,
    "eripTrxId":"12345679",
    "info": ИООО \"Ваша любимая компания\",
    "products":[
      {
        "invItemId":"-",
        "desc":"За услуги связи",
        "count":1,
        "amt":100000
      },
      {
        "invItemId":"-",
        "desc":"За доп услуги",
        "count":1,
        "amt":2000
      }
    ]
  }
}

```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, получение информации по ранее выставленному счету и завершение сеанса.

C#

```

CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rd</pwd>" +
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );

// Получаем информацию о счете

```

```
res = Request(
    "https://www.hutkigrosh.by/API/v1/Invoicing/Bill(4000000528202700)",
    "GET", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
    "POST", "application/xml", "", ref authCookie );
```

PHP

```
$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rd</pwd>".
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login : ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Получаем информацию о счете
$url = "https://www.hutkigrosh.by/API/v1/Invoicing/Bill(4000000510336500)";
echo "Get Bill : ".sendToHG( $req, $url, "", $cookies, HTTP_METH_GET )."<br/>";

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/LogOut";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";
```

МЕТОД BILLQRCODE({BILLID}, {RESULTTYPE}, {STRWIDTH}, {STRHEIGHT})

HTTP verb: **GET**

ОПИСАНИЕ

Формирует qr-код по выставленному счету или возвращает данные для его формирования.

ВЫЗОВ

[https://www.hutkigrosh.by/API/v1/Invoicing/BillQRCode\({billId},{resultType},{strwidth},{strheight}\)](https://www.hutkigrosh.by/API/v1/Invoicing/BillQRCode({billId},{resultType},{strwidth},{strheight}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При вызове данного метода необходимо указать внутренний идентификатор счета, заменив параметр **{billId}** на нужное значение. Также, необходимо задать тип возвращаемого результата **{resultType}**.

Возможные варианты значений:

- **image**. Будет возвращена сформированная картинка qr-кода в PNG-формате в виде массива байт.
- **text**. Будет возвращена строка, состоящая из данных, необходимых для формирования корректного qr-кода.

В случае, если в поле **resultType** будет задано значение, отличное от возможных вариантов, то результатом будет картинка qr-кода.

В параметры **{strWidth}** и **{strHeight}** необходимо передать необходимые ширину и высоту требуемого изображения.

Необходимо указать в запросе параметр **content-type** для определения вида возвращаемого ответа.

Пример запроса: [https://www.hutkigrosh.by/API/v1/Invoicing/BillQRCode\(1331812483683981092,0,174,386\)](https://www.hutkigrosh.by/API/v1/Invoicing/BillQRCode(1331812483683981092,0,174,386))

ОТВЕТ

Ответом является структура **BillQRCode**, описание полей структуры находится в разделе "Структуры данных и перечисления".

XML

```
<BillQRCode xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>500</status>
  <img>NULL</img>
</BillQRCode>
```

Json

```
{
  "status":0,
  "img": [12, 25, 34..., 32]
}
```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, получение информации по ранее выставленному счету и завершение сеанса.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
               "<user>user@org.com</user>" +
               "<pwd>paSSwo_rd</pwd>" +
               "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
                    "POST", "application/xml", creds, ref authCookie );

// Получаем информацию о счете
res = Request(
    "https://www.hutkigrosh.by/API/v1/Invoicing/
BillQRCode(1331812483683981092,0,174,386)",
    "GET", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
              "POST", "application/xml", "", ref authCookie );
```

PHP

```
$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
          "<user>user@org.com</user>".
          "<pwd>paSSwo_rd</pwd>".
```

```
        "</Credentials>";

        // Начинаем сеанс, если в ответ приходит false, значит аутентификация
        // провалилась и дальнейшие вызовы бессмысленные
        echo "Login : ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
        $cookies = $req -> get_cookies();

        // Получаем информацию о счете
        $url = "https://www.hutkigrosh.by/API/v1/Invoicing/
        BillQRCode(1331812483683981092,0,174,386)";
        echo "Get BillQRCode : ".sendToHG( $req, $url, "", $cookies, HTTP_METH_GET
        )."<br/>";

        // Завершаем сеанс
        $url = "https://www.hutkigrosh.by/API/v1/Security/Logout";
        echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";
```

МЕТОД BILL({BILLID})

HTTP verb: **DELETE**

ОПИСАНИЕ

Удаляет выставленный счет из системы.

ВЫЗОВ

[https://www.hutkigrosh.by/API/v1/Invoicing/Bill\({billId}\)](https://www.hutkigrosh.by/API/v1/Invoicing/Bill({billId}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При вызове данного метода необходимо указать внутренний идентификатор счета, заменив параметр **{billId}** на нужное значение. Необходимо указать в запросе параметр **content-type** для определения вида возвращаемого ответа.

Пример запроса: [https://www.hutkigrosh.by/API/v1/Invoicing/Bill\(4000000528202700\)](https://www.hutkigrosh.by/API/v1/Invoicing/Bill(4000000528202700))

ОТВЕТ

Ответом является структура **BillStatus**, описание полей структуры находится в разделе "Структуры данных и перечисления".

XML

```
<BillStatus xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>3221291522</status>
  <purchItemStatus>Payed</purchItemStatus>
</BillStatus>
```

Json

```
{
  "status":3221291522,
  "purchItemStatus":3
}
```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, удаление ранее выставленного счета и завершение сеанса.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rd</pwd>" +
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );

// Удаляем выставленный счет из системы, если счет оплачен, то его удалить
// нет возможности
res = Request(
    "https://www.hutkigrosh.by/API/v1/Invoicing/Bill(4000000528202700)",
    "DELETE", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
    "POST", "application/xml", "", ref authCookie );
```

PHP

```
$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rd</pwd>".
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login : ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Удаляем выставленный счет из системы, если счет оплачен, то его удалить
// нет возможности
$url = "https://www.hutkigrosh.by/API/v1/Invoicing/Bill(4000000510336500)";
echo "Delete: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_DELETE )."<br/>";

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/LogOut";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";
```

МЕТОД BILLS({STARTPOSITION},{REQUESTITEMSCOUNT},{SORTTYPE})

HTTP verb: GET

ОПИСАНИЕ

Возвращает список выставленных счетов. В этот список не попадают счета, которые были оплачены, и по ним пришло окончательное подтверждение оплаты. Окончательное подтверждение оплаты – это процедура, происходящая один раз в день по сверке уже имеющейся информации об оплате счетов с подтвержденной банками фактическими оплатами. После этой процедуры все подтвержденные оплаченные счета переходят в архив операций.

[https://www.hutkigrosh.by/API/v1/Invoicing/Bills\({startPosition},{requestItemsCount},{sortType}\)](https://www.hutkigrosh.by/API/v1/Invoicing/Bills({startPosition},{requestItemsCount},{sortType}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При вызове данного метода необходимо указать номер записи с которой возвращаются записи, заменив параметр **{startPosition}** на номер записи в списке с которого начнется построение результирующего набора данных, **{requestItemsCount}** этот параметр надо заменить на количество записей в результирующем наборе, **{sortType}** на один из вариантов сортировки результирующего набора. Варианты сортировки представлены в таблице:

{sortType}	Описание
1	По возрастанию номера счета
2	По убыванию номера счета
3	По возрастанию даты оплаты
4	По убыванию даты оплаты
4	По возрастанию суммы счета
6	По убыванию суммы счета
7	По возрастанию статуса счета
8	По убыванию статуса счета
9	По возрастанию данных в поле плательщик
10	По убыванию данных в поле плательщик

Необходимо указать в запросе параметр **content-type** для определения вида возвращаемого ответа.

Пример запроса: [https://www.hutkigrosh.by/API/v1/Invoicing/Bills\(0,30,1\)](https://www.hutkigrosh.by/API/v1/Invoicing/Bills(0,30,1))

ОТВЕТ

Ответом является структура **BillsInfo**, описание полей структуры находится в разделе "Структуры данных и перечисления". В возвращаемой структуре в поле **count** указано общее количество счетов доступных для выгрузки.

XML

```
<BillsInfo xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>0</status>
  <itemInThisRequest>2</itemInThisRequest>
  <startPosition>0</startPosition>
  <count>2</count>
  <bill>
    <Bill>
      <billID>4000000517424000</billID>
      <eripId>0</eripId>
      <invId>4955</invId>
      <dueDt>2011-06-09T00:00:00</dueDt>
      <addedDt>2011-06-06T00:00:00</addedDt>
      <payedDt>2011-06-06T15:59:35+03:00</payedDt>
      <fullName>Иванов А Я</fullName>
      <mobilePhone>+375 (29) 6522757</mobilePhone>
      <notifyByMobilePhone>false</notifyByMobilePhone>
      <email i:nil="true"/>
      <notifyByEmail>false</notifyByEmail>
      <fullAddress i:nil="true"/>
      <amt>5900</amt>
      <curr>BYN</curr>
      <statusEnum>Payed</statusEnum>
      <eripTrxId>12345679</eripTrxId>
      <info>ИООО "Ваша любимая компания"</info>
      <products>
        <ProductInfo>
          <invItemId>1</invItemId>
        </ProductInfo>
      </products>
    </Bill>
  </bill>
</BillsInfo>
```

```

                <desc>за интернет</desc>
                <count>1</count>
                <amt>5900</amt>
            </ProductInfo>
        </products>
    </Bill>
</bill>
</BillsInfo>

```

Json

```

{
  "status":0,
  "itemInThisRequest":2,
  "startPosition":0,
  "count":2,
  "bill":[
    {
      "billID":4000000517424000,
      "eripId":0,
      "invId":"4955",
      "dueDt":"\\/Date (1307566800000+0300) \\/",
      "addedDt":"\\/Date (1307307600000+0300) \\/",
      "payedDt":"\\/Date (1307365175000+0300) \\/",
      "fullName":"Иванов А Я",
      "mobilePhone":"+375 (29) 6522757",
      "notifyByMobilePhone":false,
      "email":null,
      "notifyByEmail":false,
      "fullAddress":null,
      "amt":5900,
      "curr":"BYN",
      "statusEnum":5,
      "eripTrxId": "1234569",
      "info": "ИООО \\\"Ваша любимая компания\\\"",
      "products":[
        {
          "invItemId":"1",
          "desc":"за интернет",
          "count":1,
          "amt":5900
        }
      ]
    },
    {

```

```

    "billID":4000000518073600,
    "eripId":0,
    "invId":"1234",
    "dueDt":"\\/Date(1307566800000+0300)\\/",
    "addedDt":"\\/Date(1307307600000+0300)\\/",
    "payedDt":"\\/Date(1307365177000+0300)\\/",
    "fullName":"тест",
    "mobilePhone":"+375 29 3333333",
    "notifyByMobilePhone":false,
    "email":null,
    "notifyByEmail":false,
    "fullAddress":null,
    "amt":10000,
    "curr":"BYN",
    "statusEnum":5,
    "eripTrxId": "1234569",
    "info": ИООО \\"Ваша любимая компания\"",
    "products":[
      {
        "invItemId":"1",
        "desc":"за интернет",
        "count":1,
        "amt":10000
      }
    ]
  }
}

```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, получение списка выставленных счетов в системе и завершение сеанса.

C#

```

CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rd</pwd>" +
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );

// Получаем список счетов
res = Request(
    "https://www.hutkigrosh.by/API/v1/Invoicing/Bills(0,30,1)",
    "GET", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
    "POST", "application/xml", "", ref authCookie );

```

PHP

```

$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rd</pwd>".
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация

```



```
// провалилась и дальнейшие вызовы бессмысленные
echo "Login: ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Получаем список счетов
$url = "https://www.hutkigrosh.by/API/v1/Invoicing/Bills(0,30,1)";
echo "Get Bills: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_GET )."<br/>";

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/Logout";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";
```

МЕТОД BILLSEX({STARTDT},{ENDDT},{STARTPOSITION},{REQUESTITEMSCOUNT}, {SORTTYPE},{SPNAME},{PURCHITEMSTATUS})

HTTP verb: **GET**

ОПИСАНИЕ

Возвращает список выставленных счетов. Это более гибкая форма получения списка счетов, так как здесь имеется возможность явно указать из какой базы нужно получить список, указать интервал дат и состояние счета.

ВЫЗОВ

[https://www.hutkigrosh.by/API/v1/Invoicing/BillsEx\({startDt},{endDt},{startPosition},{requestItemsCount},{sortType},{spName},{purchaseStatus}\)](https://www.hutkigrosh.by/API/v1/Invoicing/BillsEx({startDt},{endDt},{startPosition},{requestItemsCount},{sortType},{spName},{purchaseStatus}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При вызове данного метода необходимо указать интервал дат, указав начальную и конечную дату в результирующем списке, заменив параметры **{startDt}** и **{endDt}**. Так же номер записи, с которой возвращаются записи, заменив параметр **{startPosition}** на номер записи в списке с которого начнется построение результирующего набора данных. **{requestItemsCount}** этот параметр надо заменить на количество записей в результирующем наборе. **{sortType}** на один из вариантов сортировки результирующего набора. **{spName}** указывает на источник, откуда будет построен список, а **{purchaseStatus}** ограничивает результат списком счетов заданного статуса. Варианты сортировки представлены в таблице:

{sortType}	Описание
1	По возрастанию номера счета
2	По убыванию номера счета
3	По возрастанию даты оплаты
4	По убыванию даты оплаты
4	По возрастанию суммы счета
6	По убыванию суммы счета
7	По возрастанию статуса счета
8	По убыванию статуса счета
9	По возрастанию данных в поле плательщик
10	По убыванию данных в поле плательщик

Статусы счетов находится в разделе "Структуры данных и перечисления" в таблице "Список статусов счета", необходимо в запрос подставлять числовое значение. Источник данных **{spName}** может принимать следующие значения:

{spName}	Описание
billing	Оперативная база счетов

Необходимо указать в запросе параметр **content-type** для определения вида возвращаемого ответа.

Пример запроса:

[https://www.hutkigrosh.by/API/v1/Invoicing/BillsEx\(2011-10-01,2011-12-01,0,100,1,history,2\)](https://www.hutkigrosh.by/API/v1/Invoicing/BillsEx(2011-10-01,2011-12-01,0,100,1,history,2))

ОТВЕТ

Ответом является структура **BillsInfo**, описание полей структуры находится в разделе "Структуры данных и перечисления". В возвращаемой структуре в поле **count** указано общее количество счетов доступных для выгрузки.

XML

```
<BillsInfo xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>0</status>
  <itemInThisRequest>2</itemInThisRequest>
  <startPosition>0</startPosition>
  <count>2</count>
  <bill>
    <Bill>
      <billID>4000000517424000</billID>
      <eripId>0</eripId>
      <invId>4955</invId>
      <dueDt>2011-06-09T00:00:00</dueDt>
      <addedDt>2011-06-06T00:00:00</addedDt>
      <payedDt>2011-06-06T15:59:35+03:00</payedDt>
      <fullName>Иванов А Я</fullName>
      <mobilePhone>+375 (29) 6522757</mobilePhone>
      <notifyByMobilePhone>false</notifyByMobilePhone>
      <email i:nil="true"/>
      <notifyByEmail>false</notifyByEmail>
      <fullAddress i:nil="true"/>
      <amt>5900</amt>
      <curr>BYN</curr>
      <statusEnum>Payed</statusEnum>
      <eripTrxId>12345679</eripTrxId>
      <info>ИООО "Ваша любимая компания"</info>
      <products>
        <ProductInfo>
          <invItemId>1</invItemId>
          <desc>за интернет</desc>
          <count>1</count>
          <amt>5900</amt>
        </ProductInfo>
      </products>
    </Bill>
    <Bill>
      <billID>4000000518073600</billID>
      <eripId>0</eripId>
      <invId>1234</invId>
      <dueDt>2011-06-09T00:00:00</dueDt>
      <addedDt>2011-06-06T00:00:00</addedDt>
      <payedDt>2011-06-06T15:59:37+03:00</payedDt>
      <fullName>рест</fullName>
      <mobilePhone>+375 29 3333333</mobilePhone>
      <notifyByMobilePhone>false</notifyByMobilePhone>
      <email i:nil="true"/>
      <notifyByEmail>false</notifyByEmail>
      <fullAddress i:nil="true"/>
      <amt>10000</amt>
      <curr>BYN</curr>
      <statusEnum>Payed</statusEnum>
      <eripTrxId>12345679</eripTrxId>
      <info>ИООО "Ваша любимая компания"</info>
      <products>
        <ProductInfo>
          <invItemId>1</invItemId>
```

```

        <desc>за интернет</desc>
        <count>1</count>
        <amt>10000</amt>
    </ProductInfo>
</products>
</Bill>
</bill>
</BillsInfo>

```

Json

```

{
  "status":0,
  "itemInThisRequest":2,
  "startPosition":0,
  "count":2,
  "bill":[
    {
      "billID":4000000517424000,
      "eripId":0,
      "invId":"4955",
      "dueDt":"\\/Date(1307566800000+0300)\\/","
      "addedDt":"\\/Date(1307307600000+0300)\\/","
      "payedDt":"\\/Date(1307365175000+0300)\\/","
      "fullName":"Иванов А Я",
      "mobilePhone":"+375(29)6522757",
      "notifyByMobilePhone":false,
      "email":null,
      "notifyByEmail":false,
      "fullAddress":null,
      "amt":5900,
      "curr":"BYN",
      "statusEnum":5,
      "eripTrxId":"12345679",
      "info": "ИООО \\Ваша любимая компания\\",
      "products":[
        {
          "invItemId":"1",
          "desc":"за интернет",
          "count":1,
          "amt":5900
        }
      ]
    },
    {
      "billID":4000000518073600,
      "eripId":0,
      "invId":"1234",
      "dueDt":"\\/Date(1307566800000+0300)\\/","
      "addedDt":"\\/Date(1307307600000+0300)\\/","
      "payedDt":"\\/Date(1307365177000+0300)\\/","
      "fullName":"тест",
      "mobilePhone":"+375 29 3333333",
      "notifyByMobilePhone":false,
      "email":null,
      "notifyByEmail":false,
      "fullAddress":null,
      "amt":10000,
      "curr":"BYN",
      "statusEnum":5,
      "eripTrxId":"12345679",
      "info": "ИООО \\Ваша любимая компания\\",
      "products":[
        {
          "invItemId":"1",
          "desc":"за интернет",
          "count":1,
          "amt":10000
        }
      ]
    }
  ]
}

```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, получение списка выставленных счетов в системе и завершение сеанса.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rd</pwd>" +
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );

// Получаем список счетов
res = Request(
    "https://www.hutkigrosh.by/API/v1/Invoicing/BillsEx(2011-10-01," +
    "2011-12-01,0,100,1,history,2)",
    "GET", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
    "POST", "application/xml", "", ref authCookie );
```

PHP

```
$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rd</pwd>".
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login: ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> getCookies();

// Получаем список счетов
$url = "https://www.hutkigrosh.by/API/v1/Invoicing/BillsEx(2011-10-01".
    ",2011-12-01,0,100,1,history,2)";
echo "Get Bills: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_GET )."<br/>";

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/LogOut";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";
```

МЕТОД BILLSTATUS({BILLID})

HTTP verb: GET

ОПИСАНИЕ

Возвращает статус указанного счета.

ВЫЗОВ

[https://www.hutkigrosh.by/API/v1/Invoicing/BillStatus\({billId}\)](https://www.hutkigrosh.by/API/v1/Invoicing/BillStatus({billId}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При вызове данного метода необходимо указать внутренний идентификатор счета, заменив параметр **{billId}** на нужное значение. Необходимо указать в запросе параметр **content-type** для определения вида возвращаемого ответа.

Пример запроса: [https://www.hutkigrosh.by/API/v1/Invoicing/BillStatus\(4000000528202700\)](https://www.hutkigrosh.by/API/v1/Invoicing/BillStatus(4000000528202700))

ОТВЕТ

Ответом является структура **BillStatus**, описание полей структуры находится в разделе "Структуры данных и перечисления".

XML

```
<BillStatus xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>0</status>
  <purchItemStatus>Payed</purchItemStatus>
</BillStatus>
```

Json

```
{
  "status":0,
  "purchItemStatus":3
}
```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, получение статуса оплаты выставленного счета и завершение сеанса.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
               "<user>user@org.com</user>" +
               "<pwd>paSSwo_rd</pwd>" +
               "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
                    "POST", "application/xml", creds, ref authCookie );

// Получаем статус счета
res = Request(
    "https://www.hutkigrosh.by/API/v1/Invoicing/BillStatus(4000000528202700)",
    "GET", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
              "POST", "application/xml", "", ref authCookie );
```

PHP

```

$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
           "<user>user@org.com</user>".
           "<pwd>paSSwo_rd</pwd>".
           "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login: ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Получаем статус счета
$url =
    "https://www.hutkigrosh.by/API/v1/Invoicing/BillStatus(4000000525584600)";
echo "Bill Status: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_GET
    )."<br/>";

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/LogOut";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";

```

МЕТОД PAYEDBILLS({ERIPID},{ORDER})

HTTP verb: **GET**

ОПИСАНИЕ

Возвращает список оплаченных счетов. В этот список попадают счета, которые были уже оплачены, причем не важно находятся счета в оперативной базе или в архиве.

ВЫЗОВ

[https://www.hutkigrosh.by/API/v1/Invoicing/PayedBills\({eripId},{order}\)](https://www.hutkigrosh.by/API/v1/Invoicing/PayedBills({eripId},{order}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При вызове данного метода необходимо указать идентификатор услуги в ЕРИП, заменив параметр **{eripId}**. **{order}** этот параметр надо заменить на внутренний идентификатор счета из последней записи предыдущего вызова данного метода. Если вызов метода происходит первый раз, то надо указать 0. Необходимо указать в запросе параметр **content-type** для определения вида возвращаемого ответа.

Пример запроса: [https://www.hutkigrosh.by/API/v1/Invoicing/PayedBills\(40000001,0\)](https://www.hutkigrosh.by/API/v1/Invoicing/PayedBills(40000001,0))

ОТВЕТ

Ответом является структура **BillsInfo**, описание полей структуры находится в разделе "Структуры данных и перечисления". В возвращаемой структуре в поле **count** и **startPosition** всегда указано значение -1.

XML

```

<BillsInfo xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>0</status>
  <itemInThisRequest>2</itemInThisRequest>
  <startPosition>-1</startPosition>
  <count>-1</count>
  <bill>
    <Bill>
      <billID>4000000517424000</billID>
      <eripId>0</eripId>
    </Bill>
  </bill>
</BillsInfo>

```

```

<invId>4955</invId>
<dueDt>2011-06-09T00:00:00</dueDt>
<addedDt>2011-06-06T00:00:00</addedDt>
<payedDt>2011-06-06T15:59:35+03:00</payedDt>
<fullName>Иванов А Я</fullName>
<mobilePhone>+375 (29) 6522757</mobilePhone>
<notifyByMobilePhone>>false</notifyByMobilePhone>
<email i:nil="true"/>
<notifyByEmail>>false</notifyByEmail>
<fullAddress i:nil="true"/>
<amt>5900</amt>
<curr>BYN</curr>
<statusEnum>Payed</statusEnum>
<eripTrxId>1234569</eripTrxId>
<info>ИООО "Ваша любимая компания"</info>
<products>
  <ProductInfo>
    <invItemId>1</invItemId>
    <desc>за интернет</desc>
    <count>1</count>
    <amt>5900</amt>
  </ProductInfo>
</products>
</Bill>
<Bill>
  <billID>4000000518073600</billID>
  <eripId>0</eripId>
  <invId>1234</invId>
  <dueDt>2011-06-09T00:00:00</dueDt>
  <addedDt>2011-06-06T00:00:00</addedDt>
  <payedDt>2011-06-06T15:59:37+03:00</payedDt>
  <fullName>рест</fullName>
  <mobilePhone>+375 29 3333333</mobilePhone>
  <notifyByMobilePhone>>false</notifyByMobilePhone>
  <email i:nil="true"/>
  <notifyByEmail>>false</notifyByEmail>
  <fullAddress i:nil="true"/>
  <amt>10000</amt>
  <curr>BYN</curr>
  <statusEnum>Payed</statusEnum>
  <eripTrxId>1234569</eripTrxId>
  <info>ИООО "Ваша любимая компания"</info>
  <products>
    <ProductInfo>
      <invItemId>1</invItemId>
      <desc>за интернет</desc>
      <count>1</count>
      <amt>10000</amt>
    </ProductInfo>
  </products>
</Bill>
</bill>
</BillsInfo>

```

Json

```

{
  "status":0,
  "itemInThisRequest":2,
  "startPosition":-1,
  "count":-1,
  "bill":[
    {
      "billID":4000000517424000,
      "eripId":0,
      "invId":"4955",
      "dueDt":"\\/Date(1307566800000+0300)\\/\"",
      "addedDt":"\\/Date(1307307600000+0300)\\/\"",
      "payedDt":"\\/Date(1307365175000+0300)\\/\"",
      "fullName":"Иванов А Я",
      "mobilePhone":"+375 (29) 6522757",

```

```

        "notifyByMobilePhone":false,
        "email":null,
        "notifyByEmail":false,
        "fullAddress":null,
        "amt":5900,
        "curr":"BYN",
        "statusEnum":5,
        "eripTrxId":"12345679",
        "info": ИООО \Ваша любимая компания\",
        "products":[
            {
                "invItemId":"1",
                "desc":"за интернет",
                "count":1,
                "amt":5900
            }
        ]
    },
    {
        "billID":4000000518073600,
        "eripId":0,
        "invId":"1234",
        "dueDt":"\/Date(1307566800000+0300)\/",
        "addedDt":"\/Date(1307307600000+0300)\/",
        "payedDt":"\/Date(1307365177000+0300)\/",
        "fullName":"тест",
        "mobilePhone":"+375 29 3333333",
        "notifyByMobilePhone":false,
        "email":null,
        "notifyByEmail":false,
        "fullAddress":null,
        "amt":10000,
        "curr":"BYN",
        "statusEnum":5,
        "eripTrxId":"12345679",
        "info": ИООО \Ваша любимая компания\",
        "products":[
            {
                "invItemId":"1",
                "desc":"за интернет",
                "count":1,
                "amt":10000
            }
        ]
    }
}

```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, получение списка оплаченных счетов в системе и завершение сеанса.

C#

```

CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rd</pwd>" +
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );

// Получаем список счетов
res = Request(
    "https://www.hutkigrosh.by/API/v1/Invoicing/PayedBills(40000001,0)",
    "GET", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

```



```
// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
              "POST", "application/xml", "", ref authCookie );
```

PHP

```
$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
           "<user>user@org.com</user>".
           "<pwd>paSSwo_rd</pwd>".
           "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login: ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Получаем список счетов
$url = "https://www.hutkigrosh.by/API/v1/Invoicing/PayedBills(40000001,0)";
echo "Payed Bills: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_GET
                               )."<br/>";

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/LogOut";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";
```

МЕТОД `BILLSINMEMORIALSLIP({MEMSLIP})`

HTTP verb: **GET**

ОПИСАНИЕ

Возвращает список счетов, включенных в мемориальный ордер.

ВЫЗОВ

[https://www.hutkigrosh.by/API/v1/Invoicing/BillsInMemorialSlip\({memSlip}\)](https://www.hutkigrosh.by/API/v1/Invoicing/BillsInMemorialSlip({memSlip}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При вызове данного метода необходимо указать номер мемориального ордера, заменив параметр `{memSlip}`. Необходимо указать в запросе параметр `content-type` для определения вида возвращаемого ответа.

Пример запроса: [https://www.hutkigrosh.by/API/v1/Invoicing/BillsInMemorialSlip\(1234567\)](https://www.hutkigrosh.by/API/v1/Invoicing/BillsInMemorialSlip(1234567))

ОТВЕТ

Ответом является структура **BillsInfo**, описание полей структуры находится в разделе "Структуры данных и перечисления". В возвращаемой структуре в поле `count` и `startPosition` всегда указано значение -1.

XML

```
<BillsInfo xmlns="http://www.hutkigrosh.by/api/invoicing">
  <status>0</status>
  <itemInThisRequest>2</itemInThisRequest>
  <startPosition>-1</startPosition>
  <count>-1</count>
```

```

<bill>
  <Bill>
    <billID>4000000517424000</billID>
    <eripId>0</eripId>
    <invId>4955</invId>
    <dueDt>2011-06-09T00:00:00</dueDt>
    <addedDt>2011-06-06T00:00:00</addedDt>
    <payedDt>2011-06-06T15:59:35+03:00</payedDt>
    <fullName>Иванов А Я</fullName>
    <mobilePhone>+375 (29) 6522757</mobilePhone>
    <notifyByMobilePhone>false</notifyByMobilePhone>
    <email i:nil="true"/>
    <notifyByEmail>false</notifyByEmail>
    <fullAddress i:nil="true"/>
    <amt>5900</amt>
    <curr>BYN</curr>
    <statusEnum>Paid</statusEnum>
    <eripTrxId>1234698</eripTrxId>
    <info>ЮОО "Ваша любимая компания"</info>
    <products>
      <ProductInfo>
        <invItemId>1</invItemId>
        <desc>за интернет</desc>
        <count>1</count>
        <amt>5900</amt>
      </ProductInfo>
    </products>
  </Bill>
  <Bill>
    <billID>4000000518073600</billID>
    <eripId>0</eripId>
    <invId>1234</invId>
    <dueDt>2011-06-09T00:00:00</dueDt>
    <addedDt>2011-06-06T00:00:00</addedDt>
    <payedDt>2011-06-06T15:59:37+03:00</payedDt>
    <fullName>рест</fullName>
    <mobilePhone>+375 29 3333333</mobilePhone>
    <notifyByMobilePhone>false</notifyByMobilePhone>
    <email i:nil="true"/>
    <notifyByEmail>false</notifyByEmail>
    <fullAddress i:nil="true"/>
    <amt>10000</amt>
    <curr>BYN</curr>
    <statusEnum>Paid</statusEnum>
    <eripTrxId>1234698</eripTrxId>
    <info>ЮОО "Ваша любимая компания"</info>
    <products>
      <ProductInfo>
        <invItemId>1</invItemId>
        <desc>за интернет</desc>
        <count>1</count>
        <amt>10000</amt>
      </ProductInfo>
    </products>
  </Bill>
</bill>
</BillsInfo>

```

Json

```

{
  "status":0,
  "itemInThisRequest":2,
  "startPosition":-1,
  "count":-1,
  "bill":[
    {
      "billID":4000000517424000,
      "eripId":0,
      "invId":"4955",
      "dueDt":"\\/Date(1307566800000+0300)\\/",

```

```

        "addedDt": "\/Date(1307307600000+0300)\/",
        "payedDt": "\/Date(1307365175000+0300)\/",
        "fullName": "Иванов А Я",
        "mobilePhone": "+375 (29) 6522757",
        "notifyByMobilePhone": false,
        "email": null,
        "notifyByEmail": false,
        "fullAddress": null,
        "amt": 5900,
        "curr": "BYN",
        "statusEnum": 5,
        "eripTrxId": "1234569",
        "info": "ИООО \"/>

```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом, получение списка оплаченных счетов в системе и завершение сеанса.

C#

```

CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rdc/pwd</pwd>" +
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );

// Получаем список счетов

```

```

res = Request(
    "https://www.hutkigrosh.by/API/v1/Invoicing/BillsInMemorialSlip(1234567)",
    "GET", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/Logout",
    "POST", "application/xml", "", ref authCookie );

```

PHP

```

$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns='http://www.hutkigrosh.by/api/'>".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rd</pwd>".
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login: ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Получаем список счетов
$url = "https://www.hutkigrosh.by/API/v1/Invoicing/BillsInMemorialSlip(123456)";
echo "Payed Bills: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_GET
    )."<br/>";

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/Logout";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";

```

ПОДСИСТЕМА ОПЛАТЫ

<https://www.hutkigrosh.by/API/v1/Pay>

Метод		Описание
WebPay	POST	Возвращает форму виджета для перехода к оплате картой на сайт WebPay
AlfaClick	POST	Выставляет счет в AlfaClick, для заданного клиента

МЕТОД WEBPAY

HTTP verb: **POST**

ОПИСАНИЕ

Метод используется для получения формы виджета оплаты счета с помощью банковской карты. Алгоритм действий таков: Вы выставляете счет, получаете **billId**, далее, используя этот запрос, Вы получаете **html** код для вставки на Вашу страницу, где будет небольшая форма и кнопка для перехода на сервер оплаты.

ВЫЗОВ

<https://www.hutkigrosh.by/API/v1/Pay/WebPay>

Для вызова данного метода необходимо сформировать тело запроса. Запрос представляет собой заполненную структуру **WebPayParam**, описание полей структуры находится в разделе "Структуры данных и перечисления". В структуре информация об оплачиваемом счете может находиться или в поле **billId** или в структуре **orderData**, если заполнены одновременно оба поля, то приоритет имеет поле **billId**. Тело запроса зависит от выбранного значения **content-type**:

XML

```
<WebPayParam xmlns="http://www.hutkigrosh.by/API/PaymentSystems">
  <billId />
  <orderData>
    <serviceProviderId>12000</serviceProviderId >
    <eripId>12000001</eripId >
    <spClaimId>CP-1234</spClaimId>
    <amount>10000</amount>
    <currency>933</currency>
    <clientFio>Пупкин Иван И.</clientFio>
    <clientAddress>Город, улица, дом</clientAddress>
    <trxId>23442882</trxId>
  </orderData>
  <returnUrl>http://localhost/?success</returnUrl>
  <cancelReturnUrl>http://localhost/?error</cancelReturnUrl>
  <submitValue>Оплатить картой</cancelReturnUrl>
</WebPayParam>
```

Json

```
{
  "billId":null,
  "orderData":{
    "serviceProviderId":12000,
    "eripId":12000001,
    "spClaimId":"CP-1234",
    "amount":"10000",
    "currency":"933",
    "clientFio":" Пупкин Иван И.",
    "clientAddress":" Город, улица, дом ",
    "trxId":"23442882"
  },
  "returnUrl":"http://localhost/?success",
  "cancelReturnUrl":"http://localhost/?error",
  "submitValue":"Оплатить картой",
}
```

ОТВЕТ

Ответом является структура **WebPayForm**, в которой находится **html** форма для отображения на странице сайта.

МЕТОД ALFAClick

HTTP verb: **POST**

ОПИСАНИЕ

Метод используется для выставления счета в систему **AlfaClick**.

ВЫЗОВ

<https://www.hutkigrosh.by/API/v1/Pay/AlfaClick>

Для вызова данного метода необходимо сформировать тело запроса. Запрос представляет собой заполненную структуру **AlfaClickParam**, описание полей структуры находится в разделе "Структуры данных и перечисления". В структуре информация об оплачиваемом счете находится в поле **billId**. Так же обязательным является номер телефона, который указывает клиента, которому будет выставлен счет в системе **AlfaClick**. Тело запроса зависит от выбранного значения **content-type**:

XML

```
<AlfaClickParam xmlns="http://www.hutkigrosh.by/API/PaymentSystems">
  <billId>3248979861349</billId>
  <phone>375291234567</phone>
</AlfaClickParam>
```

Json

```
{
  "billId": "3248979861349",
  "phone": "375291234567"
}
```

ОТВЕТ

Ответ приходит **0** – если произошла ошибка, **billId** – если удалось выставить счет в **AlfaClick**.

XML

```
<long
xmlns="http://schemas.microsoft.com/2003/10/Serialization/">9223372036854775807</long>
```

Json

```
9223372036854775807
```

ПОДСИСТЕМА РАБОТЫ С КУРСАМИ ВАЛЮТ

<https://www.hutkigrosh.by/API/v1/Currency>

Метод		Описание
Currency({isArchive})	GET	Возвращает установленные курсы или архив курсов, в зависимости от параметра
Currency	POST	Устанавливает курсы

МЕТОД CURRENCY({ISARCHIVE})

HTTP verb: **GET**

ОПИСАНИЕ

Возвращает действующие курсы для пересчета в базовую валюту (в нашем случае это BYN) или архив изменений курсов валют.

[https://www.hutkigrosh.by/API/v1/Currency/Currency\({isArchive}\)](https://www.hutkigrosh.by/API/v1/Currency/Currency({isArchive}))

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

При формировании запроса необходимо указать источник получения курсов в параметре **{isArchive}**. Значение данного параметра должно быть равным **1** – если необходимы архивные данные, или **0** – если необходимы действующие курсы валют.

Пример запроса: [http://www.hutkigrosh.by/API/v1/Currency/Currency\(0\)](http://www.hutkigrosh.by/API/v1/Currency/Currency(0))

ОТВЕТ

Ответом является заполненная структура **CurrencyInfo**, описание полей структуры находится в разделе "Структуры данных и перечисления".

XML

```
<CurrencyInfo xmlns="http://www.hutkigrosh.by/api/currency">
  <status>0</status>
  <currency>
    <convertRule>General</convertRule>
    <rate>0</rate>
    <currRate>
      <ExchangeRateInfo>
        <currCode>840</currCode>
        <rate>5230</rate>
        <endDt>1999-05-31T11:20:00</endDt>
        <startDt>1999-05-30T11:20:00</startDt>
      </ExchangeRateInfo>
      <ExchangeRateInfo>
        <currCode>643</currCode>
        <rate>270</rate>
        <endDt>1999-05-31T11:20:00</endDt>
        <startDt>1999-05-30T11:20:00</startDt>
      </ExchangeRateInfo>
    </currRate>
  </currency>
</CurrencyInfo>
```

Json

```
{
  "status":0,
  "currency":{
    "convertRule":0,
    "rate":0,
    "currRate":[{
      "currCode":"840",
      "rate":5230,
      "endDt":"\\/Date(928138800000+0300)\\/\"",
      "startDt":"\\/Date(928138800000+0300)\\/\""
    }]
  }
}
```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом. Получение списка действующих курсов и завершение сеанса.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rd</pwd>" +
    "</Credentials>";
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );

// Получаем список действующих курсов валют
res = Request(
    "https://www.hutkigrosh.by/API/v1/Currency/Currency(0)",
    "GET", "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
    "POST", "application/xml", "", ref authCookie );
```

PHP

```
$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rd</pwd>".
    "</Credentials>";

echo "Login : ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Получаем список действующих курсов валют
$url = "https://www.hutkigrosh.by/API/v1/Currency/Currency(0)";
echo "Currencies: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_GET
    )."<br/>";

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/LogOut";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";
```

МЕТОД CURRENCY

HTTP verb: **POST**

ОПИСАНИЕ

Метод устанавливает курсы валют.

ВЫЗОВ

<https://www.hutkigrosh.by/API/v1/Currency/Currency>

ПАРАМЕТРЫ ИЛИ ТЕЛО ЗАПРОСА

Для вызова данного метода необходимо сформировать тело запроса. Запрос представляет собой заполненную структуру **Currency**, описание полей структуры находится в разделе "Структуры данных и перечисления". В зависимости от содержимого поля **convertRule** необходимо заполнять различные части структуры:

convertRule	Описание
General или 1	Этот режим указывает, что будут использованы установленные НБ РБ курсы валют. Остальные поля структуры можно не заполнять.
CustomerProvided или 2	В этом режиме используются курсы валют, которые задает поставщик услуг. Блок currRate должен быть заполнен, но в этом блоке нет необходимости заполнять поля startDt и endDt , так как они используются только при переходе курса в архив, а выставаемый курс начинает действовать немедленно с времени выставления.
GeneralPercent или 3	В этом режиме используются курсы НБ РБ, но к ним применяется коэффициент, указанный в поле rate . Блок currRate можно не заполнять.

Тело запроса зависит от выбранного значения **content-type**:

XML

```
<Currency xmlns="http://www.hutkigrosh.by/api/currency">
  <convertRule>CustomerProvided</convertRule>
  <rate>0</rate>
  <currRate>
    <ExchangeRateInfo>
      <currCode>840</currCode>
      <rate>5320</rate>
      <startDt>1999-05-31T11:20:00</startDt>
    </ExchangeRateInfo>
    <ExchangeRateInfo>
      <currCode>643</currCode>
      <rate>279</rate>
      <startDt>1999-05-31T11:20:00</startDt>
    </ExchangeRateInfo>
  </currRate>
</Currency>
```

Json

```
{
  "convertRule":2,
  "rate":0,
  "currRate":[{"currCode":"840",
    "rate":5320,
    "startDt":"\\/Date(928138800000+0300)\\/"}
  ]
}
```

ОТВЕТ

В ответ приходит статус выполнения операции. Все коды состояния можно найти в разделе "Коды ошибок".

XML

```
<unsignedInt xmlns="http://schemas.microsoft.com/2003/10/Serialization/">
  0
</unsignedInt>
```

Json

```
0
```

ПРИМЕР КОДА

Запрос на аутентификацию и получение набора cookie, связанного с запросом и завершение сеанса.

C#

```
CookieContainer authCookie = new CookieContainer();
string creds = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">" +
    "<user>user@org.com</user>" +
    "<pwd>paSSwo_rdc</pwd>" +
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
string res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogIn",
    "POST", "application/xml", creds, ref authCookie );

// Устанавливаем курсы валют
res = "<Currency xmlns=\"http://www.hutkigrosh.by/api/currency\">" +
    "<convertRule>CustomerProvided</convertRule>" +
    "<rate>0</rate>" +
    "<currRate>" +
    "<ExchangeRateInfo>" +
    "<currCode>840</currCode>" +
    "<rate>5320</rate>" +
    "<startDt>1999-05-31T11:20:00</startDt>" +
    "</ExchangeRateInfo>" +
    "<ExchangeRateInfo>" +
    "<currCode>643</currCode>" +
    "<rate>279</rate>" +
    "<startDt>1999-05-31T11:20:00</startDt>" +
    "</ExchangeRateInfo>" +
    "</currRate>" +
    "</Currency>";
res = Request( "https://www.hutkigrosh.by/API/v1/Currency/Currency", "POST",
    "application/xml", res, ref authCookie );
// Теперь необходимо обработать результат, который вернул сервер

// Завершаем сеанс
res = Request( "https://www.hutkigrosh.by/API/v1/Security/LogOut",
    "POST", "application/xml", "", ref authCookie );
```

PHP

```
$url = "https://www.hutkigrosh.by/API/v1/Security/LogIn";
$req =& new HTTPRequest( $url );
$dataXML = "<Credentials xmlns=\"http://www.hutkigrosh.by/api\">".
    "<user>user@org.com</user>".
    "<pwd>paSSwo_rdc</pwd>".
    "</Credentials>";

// Начинаем сеанс, если в ответ приходит false, значит аутентификация
// провалилась и дальнейшие вызовы бессмысленные
echo "Login : ".sendToHG( $req, $url, $dataXML, null, HTTP_METH_POST )."<br/>";
$cookies = $req -> get_cookies();

// Выставляем новый счет
$url = "https://www.hutkigrosh.by/API/v1/Currency/Currency";
$dataXML =
    "<Currency xmlns=\"http://www.hutkigrosh.by/api/currency\">" +
    "<convertRule>CustomerProvided</convertRule>" +
    "<rate>0</rate>" +
    "<currRate>" +
    "<ExchangeRateInfo>" +
    "<currCode>840</currCode>" +
    "<rate>5320</rate>" +
    "<startDt>1999-05-31T11:20:00</startDt>" +
```

```

        "</ExchangeRateInfo>" +
        "<ExchangeRateInfo>" +
        "    <currCode>643</currCode>" +
        "    <rate>279</rate>" +
        "    <startDt>1999-05-31T11:20:00</startDt>" +
        "</ExchangeRateInfo>" +
        "</currRate>" +
        "</Currency>";
echo "Set Currency : ".sendToHG( $req, $url, $dataXML, $cookies,
    HTTP_METH_POST )."<br/>";
// Теперь необходимо обработать результат, который вернул сервер
// для этого надо изменить предыдущую строку соответствующим образом

// Завершаем сеанс
$url = "https://www.hutkigrosh.by/API/v1/Security/Logout";
echo "Logout: ".sendToHG( $req, $url, "", $cookies, HTTP_METH_POST )."<br/>";

```

SOAP СЕРВИС

Данный сервис служит для эмуляции вызовов, совместимых с системой EasyPay. На данный момент эмуляция ограничена единственным методом возвращающим список оплаченных счетов. Расширение списка возможно по согласованию.

Контракт сервиса расположен по адресу:

<http://www.hutkigrosh.by/API/SOAP/EasyPay.svc?wsdl>

МЕТОД EP_GETPAIDINVOICESEXT

ОПИСАНИЕ

Получение списка оплаченных счетов с подробной информацией.

ПРОТОТИП

```
Status EP_GetPaidInvoicesExt( out InvoiceExt[] invoices_ext, string mer_no, string pass,
string order )
```

ПАРАМЕТРЫ

- mer_no – Ваш логин к сервису «Хуткі Грош™»,
- pass – Ваш пароль,
- order – уникальный номер электронного счета и идентификатор услуги в ЕРИП в формате <order>|<eripId>, где <order> - номер последнего ранее полученного счета, при вызове этого метода или 0 если необходимо получить список оплаченных счетов с самого начала, <eripId> - идентификатор услуги в сервисе «Хуткі Грош™»,
- invoices_ext – список оплаченных счетов.

ОТВЕТ

В ответ приходит структура **Status**, в которой содержатся коды ответа сервера с описанием.

Поле	Тип	Описание
------	-----	----------

code	int	Код ответа сервера: 30 - неверный формат переданных данных, 31 - ошибка соединения с сервером EasyPay, 32 - неверный формат ответа сервера EasyPay, 200 - счет оплачен, 211 - счет не оплачен, 306 - идентификатор (номер) электронного кошелька EasyPay содержит недопустимые символы, 330 - сумма меньше 100 руб. Необходимо увеличить сумму для проведения операции, 401 - ошибка авторизации, 403 - нет доступа, 405 - метод не поддерживается, 500 - неизвестная внутренняя ошибка. При возврате ошибки с кодом "500" обратитесь к Администратору, 501 - несуществующий Идентификатор (номер) электронного кошелька EasyPay, 503 - такой счет не существует в системе (Покупатель может самостоятельно удалить неоплаченный счет из системы), 506 - счет просрочен, 517 - счет отменен.
message	string	комментарий на code ошибки от сервера на русском языке. Формируется сервером EasyPay, поясняет код ошибки

А так же собственно список оплаченных счетов **InvoiceExt**:

Поле	Тип	Описание
order	string	Номер счета
purch_date	DateTime	Дата и время оплаты счета
sum	decimal	Сумма в белорусских рублях
exp	DateTime	Дата окончания времени жизни счета
card	string	Идентификатор электронного кошелька EasyPay (в нашем случае пустое поле)
comment	string	Краткий комментарий счета (в нашем случае пустое поле)
info	string	Полный комментарий счета (в нашем случае пустое поле)
xml	string	Дополнительное описание счета в произвольной форме. В нашем случае это строка вида: строка вида "<ext_schet>то что вводит клиент при оплате (лицевой счет)</ext_schet><ext_money>сумма в бел.рублях</ext_money><ext_card>номер счета (равен значению из поля order)</ext_card>"

ПРИМЕР КОДА

Получение списка оплаченных счетов.

C#

```
InvoicesExt[] inv = null;
Status s = srv.EP_GetPaidInvoicesExt( out inv, "xxxxxxxx@xxxxx.xxx", "password",
"40000001|0" );
```

МЕТОД EP_VERIFYPAIDINVOICES

ОПИСАНИЕ

Проверка, подтверждены ли оплаты мемориальными ордерами.

ПРОТОТИП

```
Status EP_VerifyPaidInvoices( string mer_no, string pass, string[] orders, out VerifiedInvoice[] verifiedInvoices )
```

ПАРАМЕТРЫ

- `mer_no` – Ваш логин к сервису «Хуткі Грош™»,
- `pass` – Ваш пароль,
- `order` – массив счетов для проверки,
- `verifiedInvoices` – список счетов со статусом проверки.

ОТВЕТ

В ответ приходит структура **Status**, в которой содержатся коды ответа сервера с описанием.

Поле	Тип	Описание
<code>code</code>	<code>int</code>	Код ответа сервера: 30 - неверный формат переданных данных, 31 - ошибка соединения с сервером EasyPay, 32 - неверный формат ответа сервера EasyPay, 200 - счет оплачен, 211 - счет не оплачен, 306 - идентификатор (номер) электронного кошелька EasyPay содержит недопустимые символы, 330 - сумма меньше 100 руб. Необходимо увеличить сумму для проведения операции, 401 - ошибка авторизации, 403 - нет доступа, 405 - метод не поддерживается, 500 - неизвестная внутренняя ошибка. При возврате ошибки с кодом "500" обратитесь к Администратору, 501 - несуществующий Идентификатор (номер) электронного кошелька EasyPay, 503 - такой счет не существует в системе (Покупатель может самостоятельно удалить неоплаченный счет из системы), 506 - счет просрочен, 517 - счет отменен.
<code>message</code>	<code>string</code>	комментарий на <code>code</code> ошибки от сервера на русском языке. Формируется сервером EasyPay, поясняет код ошибки

А так же собственно список проверенных счетов **verifiedInvoices**:

Поле	Тип	Описание
<code>order</code>	<code>string</code>	Номер счета
<code>isConfirmed</code>	<code>bool</code>	Подтверждение оплаты (<code>true</code> – подтверждена)
<code>paymentSource</code>	<code>string</code>	Источник оплаты

СТРУКТУРЫ ДАННЫХ И ПЕРЕЧИСЛЕНИЯ

СТРУКТУРЫ

В описании структур используются следующие обозначения признаков:

- М – обязательное
- О – опциональное

BILLINFO

Ответ на запрос информации о выставленном счете

Название	Тип	Признак	Описание
status	uint	М	Статус ответа сервера, 0 – успешное завершение операции, иное значение – код ошибки.
bill	Bill	М	Информация о запрашиваемом счете

BILLSINFO

Список счетов

Название	Тип	Признак	Описание
status	uint	М	Статус ответа сервера, 0 – успешное завершение операции, иное значение – код ошибки.
itemInThisRequest	int	М	Количество записей в данном ответе на запрос.
itemInThisRequest	int	М	Индекс записи с которой сформирован результирующий набор.
count	long	М	Общее количество счетов на сервере.
bill	Bill[]	М	Результирующий набор счетов.

BILLSTATUS

Состояние запрашиваемого счета

Название	Тип	Признак	Описание
status	uint	М	Статус ответа сервера, 0 – успешное завершение операции, иное значение – код ошибки.
purchlItemStatus	PurchlItemStatus	М	Статус запрашиваемого счета

NEWBILLSTATUS

Результат выставления нового счета

Название	Тип	Признак	Описание
status	uint	М	Статус ответа сервера, 0 – успешное завершение операции, иное значение – код ошибки.
billID	long	М	Внутренний идентификатор выставленного счета

BILL

Информация о счете

Название	Тип	Признак	Описание
billID	long	О	Внутренний идентификатор выставленного счета
eripld	long	М	Идентификатор услуги в ЕРИП
invld	string	М	Номер счета, заказа
dueDt	DateTime	М	Срок оплаты счета
addedDt	DateTime	М	Дата выставления счета
payedDt	DateTime	О	Дата оплаты счета
fullName	string	М	ФИО клиента
mobilePhone	string	М	Мобильный телефон клиента
notifyByMobilePhone	bool	М	Состояние информирования с помощью SMS

email	string	O	Электронная почта клиента
notifyByEmail	bool	M	Состояние информирования по электронной почте
fullAddress	string	O	Полный адрес клиента
amt	Decimal	O	Сумма счета
curr	string	M	Валюта счета
firstPaymentAmt	Decimal	O	Сумма первого платежа (предоплаты) по счету
statusEnum	PurchItemStatus	M	Состояние счета
info	string	O	Владелец счета
paymentSource	string	O	Способы авторизации суммы
products	ProductInfo	M	Список товаров/услуг
eripTrxId	string	O	Идентификатор операции в ЕРИП

PRODUCTINFO

Информация о товаре или услуге

Название	Тип	Признак	Описание
invItemId	string	O	Артикул
desc	string	M	Описание товара/услуги
count	int	M	Количество
amt	Decimal	O	Стоимость за единицу

EXCHANGERATEINFO

Информация о курсе валют

Название	Тип	Признак	Описание
currCode	string	M	Код валюты в цифровом виде
rate	Decimal	M	Курс
startDt	DateTime	O	Дата начала действия
endDt	DateTime	O	Дата конца действия

CURRENCY

Курсы валют, которые устанавливает поставщик услуг

Название	Тип	Признак	Описание
convertRule	CurConvertRule	M	Режим расчета курса валют
rate	Decimal	O	Коэффициент к курсу НБ РБ
currRate	ExchangeRateInfo	O	Курсы валют

ALFACLICKPARAM

Выставление счета в систему AlfaClick

Название	Тип	Признак	Описание
billId	string	M	Внутренний идентификатор счета в системе "Хуткі Грош".
phone	string	M	Мобильный номер телефона клиента, которому будет выставлен счет

ORDERDATA

Информация о свободном счете, оплачиваемом через процессинг WebPay

Название	Тип	Признак	Описание
serviceProviderId	int	O	Идентификатор плательщика. Необходимо узнать у курирующего менеджера сервиса "Хуткі Грош"

eripld	Long	M	Идентификатор услуги. Необходимо узнать у курирующего менеджера сервиса "Хуткі Грош"
spClaimId	string	M	Идентификатор (номер) оплачиваемого счета
amount	string	O	Сумма счета к оплате. Если параметры услуги позволяют менять сумму оплаты, то к оплате будет выставлена указанная в этом поле сумма счета, если сумма счета величина неизменяемая, то это поле игнорируется и его можно не указывать.
currency	string	O	Валюта счета, если указано поле amount, то данное поле так же должно быть заполнено. Поле должно содержать код валюты в ISO формате, например, для белорусских рублей – 933.
clientFio	string	O	Фамилия плательщика
clientAddress	string	O	Адрес плательщика
trxId	string	O	Идентификатор транзакции. (В текущей реализации можно не заполнять).

WEBPAYPARAM

Параметры оплаты счета карточкой через WebPay

Название	Тип	Признак	Описание
billId	string	O	Внутренний идентификатор счета в системе "Хуткі Грош".
orderData	OrderData	O	Информация о свободном счете
returnUrl	string	M	Адрес для возврата в случае успешной оплаты счета
cancelReturnUrl	string	M	Адрес возврата в случае ошибок оплаты или отмены оплаты.
submitValue	String	M	Текст, отображаемый на кнопке для оплаты

WEBPAYFORM

Подготовленные данные для оплаты через процессинг WebPay

Название	Тип	Признак	Описание
status	uint	M	Статус ответа сервера, 0 – успешное завершение операции, иное значение – код ошибки.
form	string	M	Форма со скрытыми полями и кнопкой для совершения оплаты

BILLQRCODE

Подготовленные данные получения qr-кода для выставленного счета

Название	Тип	Признак	Описание
status	uint	M	Статус ответа сервера, 0 – успешное завершение операции, иное значение – код ошибки.
img	object	M	Сформированный результат. В случае, если status не равен 0 (ноль), поле содержит NULL.

ПЕРЕЧИСЛЕНИЯ

PURCHITEMSTATUS

Список статусов счета

Название	Код	Описание
NotSet	-1	Не установлено

PaymentPending	1	Ожидание оплаты
Outstanding	2	Просроченный
DeletedByUser	3	Удален
PaymentCancelled	4	Прерван
Payed	5	Оплачен

PURCHITEMSTATUS

Режимы пересчета счетов в валюте в BYN

Название	Код	Описание
NotSet	-1	Не установлено
General	1	Используются курсы НБ РБ
CustomerProvided	2	Используются курсы, установленные поставщиком услуг
GeneralPercent	3	Используются курсы НБ РБ помноженные на коэффициент, установленный поставщиком услуг

КОДЫ ОШИБОК

Список ошибок

Код HEX	Код DEC	Описание
0xC0010001	3221291009	Общая ошибка сервиса
0xC0010201	3221291521	Нет информации о счете
0xC0010202	3221291522	Нет возможности удалить счет
0xC0010203	3221291523	Общая ошибка выставления счета
0xC0010204	3221291524	Не указан номер счета
0xC0010205	3221291525	Счет не уникальный
0xC0010206	3221291526	Счет уже выставлен, но срок оплаты прошел
0xC0010207	3221291527	Счет выставлен и оплачен
0xC0010208	3221291528	Не указано количество товаров/услуг в заказе
0xC0010209	3221291529	Не указана сумма счета
0xC001020A	3221291530	Не указано наименование товара
0xC001020B	3221291531	Общая сумма счета меньше нуля
0xC0010251	3221291601	Возвращены не все счета
0xC0010401	3221292033	Общая ошибка установки курсов валют
0xC0010402	3221292034	Не указан коэффициент к курсу НБ РБ
0xC0010403	3221292035	Не определены курсы валют поставщика услуг
0xC0010404	3221292036	Не установлен режим пересчета курсов валют
0xC0010501	3221292289	Общая ошибка при получении курсов валют
0xC0010701	3221292801	В запросе отсутствуют параметры
0xC0010702	3221292802	Поле billId не заполнен
0xC0010703	3221292803	Нет url для перенаправления в случае успешной оплаты
0xC0010704	3221292804	Нет url для перенаправления в случае ошибки или отмены оплаты
0xC0010706	3221292806	Счет не найден в системе
0xC0010707	3221292807	Поставщик услуг не найден в системе
0xC0010708	3221292808	Услуга не полностью сконфигурирована
0xC0010709	3221292809	Нет номера услуги на платежном сайте в настройках
0xC001070A	3221292810	Счет оплачен ранее